A Globally Convergent Cutting-Plane Method for Simulation-Based Optimization with Integer Constraints

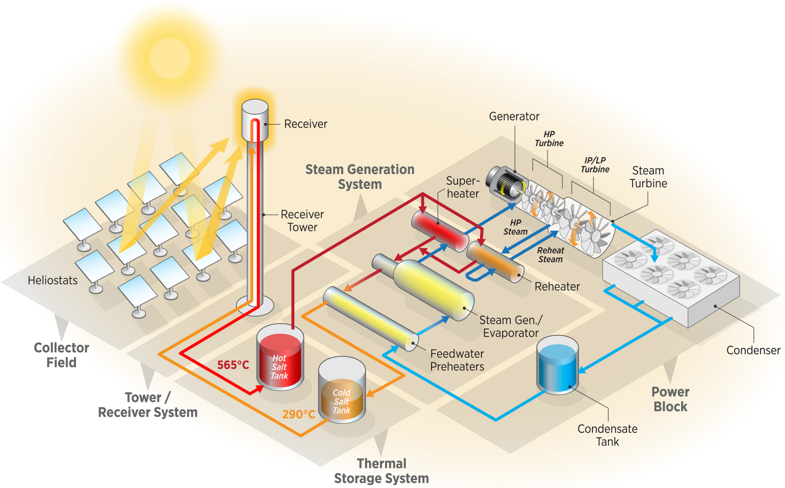
Prashant Palkar, Jeffrey Larson, Sven Leyffer, Stefan Wild

Argonne National Laboratory

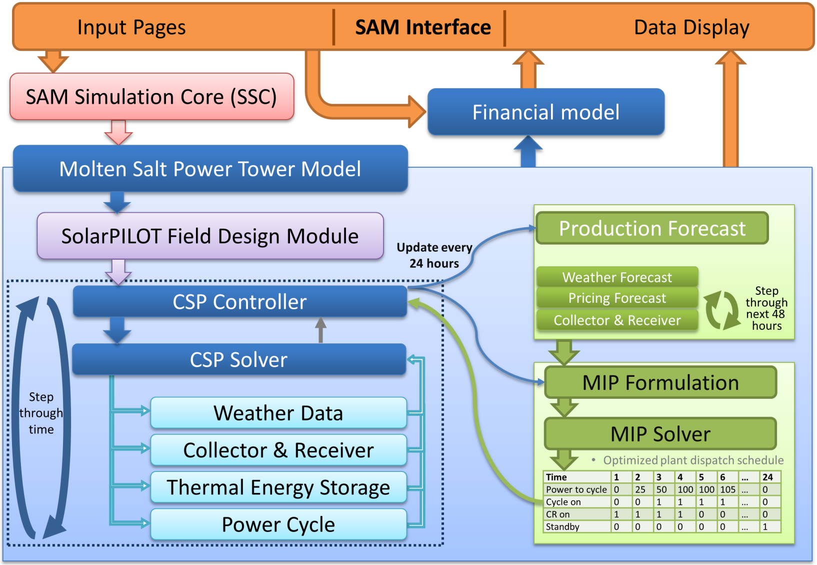
March 26, 2018

Concentrated Solar Power

Concentrated Solar Power



Concentrated Solar Power



Time	1	2	3	4	5	6	...	24
Power to cycle	0	25	50	100	100	105	...	0
Cycle on	0	0	1	1	1	1	...	0
CR on	1	1	1	1	0	0	...	0
Standby	0	0	0	0	0	0	...	1

Concentrated Solar Power

- Continuous design parameters:
  tower height, receiver diameter, receiver height, ...
- Integer design parameters:
  number of O&M staff, wash crews, and receiver panels

Unrelaxable MIP-DFO

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & \mathsf{ppaPrice}(S(x)) \\
\text{subject to} \quad & \mathsf{FluxMax}(S(x)) \leq K \\
& x \in X \subset \mathbb{Z}^q \times \mathbb{R}^p
\end{aligned}
$$

- Unrelaxable integer variables
- Computationally expensive simulation $S(x)$

Problem formulation

Derivative-Free Optimization with Unrelaxable Integers

$$\underset{x}{\text{minimize}} \; f(S(x)) \quad \text{subject to } x \in X \subset \mathbb{Z}^p \times \mathbb{R}^q$$

1. Evaluation involves $S(x)$ numerical simulation (computationally expensive)
   - derivatives $\nabla_x S$ unavailable or expensive to compute
   - single evaluation of $S(x)$ can take minutes/hours/days

   ⇒ cannot use Outer Approximation
2. Unrelaxable integers, e.g. # receiver panels
   - Unrelaxable: simulation cannot run at fractional values!

   ⇒ cannot use Branch-and-Bound

Problem formulation

Derivative-Free Optimization with Unrelaxable Integers

$$\underset{x}{\text{minimize}} \ f(S(x)) \quad \text{subject to } x \in X \subset \mathbb{Z}^p \times \mathbb{R}^q$$

1. Evaluation involves $S(x)$ numerical simulation (computationally expensive)
   - derivatives $\nabla_x S$ unavailable or expensive to compute
   - single evaluation of $S(x)$ can take minutes/hours/days

   ⇒ cannot use Outer Approximation
2. Unrelaxable integers, e.g. # receiver panels
   - Unrelaxable: simulation cannot run at fractional values!

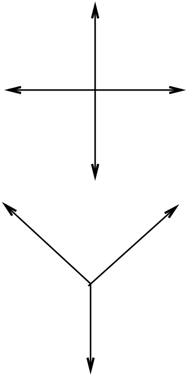
   ⇒ cannot use Branch-and-Bound

Open problem (MINLP Oberwolfach, 2015)

Solve MI-DFO for convex $f(S(x))$ without complete enumeration!

Pattern-search for continuous DFO

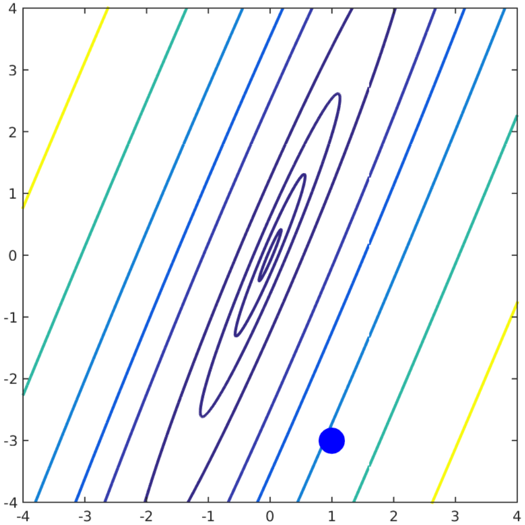
Positive spanning set of directions



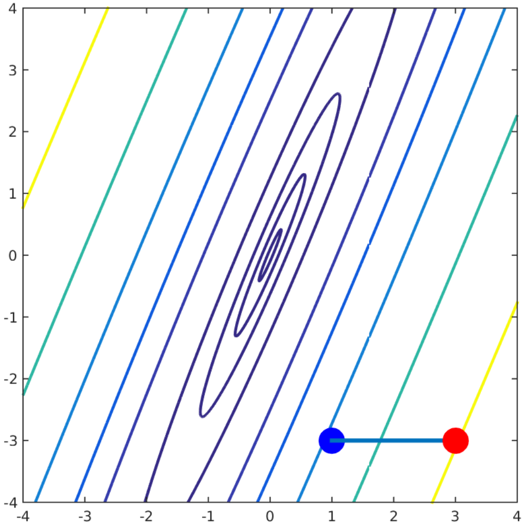
```
Select step $\Delta > 0$ and starting point $x$
repeat
    From $x$, search positive-spanning set of step $\Delta$
    if better point $\hat{x}$ found then
        Set $x := \hat{x}$; Increase $\Delta$
    else
        Decrease $\Delta$
until convergence;
```

Convergence depends on spanning set properties and smoothness of the function.

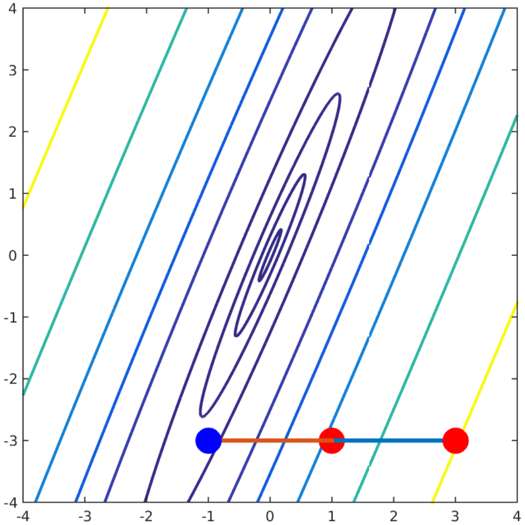
Pattern-Search Techniques for DFO



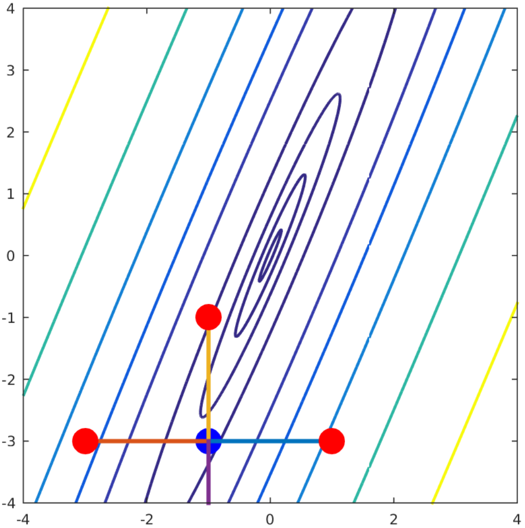
Pattern-Search Techniques for DFO



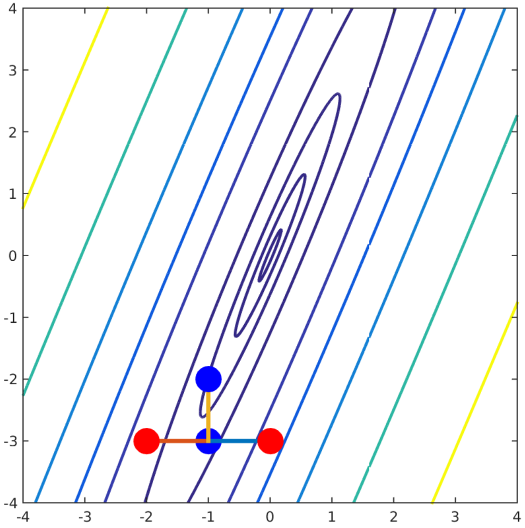
Pattern-Search Techniques for DFO



Pattern-Search Techniques for DFO



Pattern-Search Techniques for DFO



Pattern-Search Algorithm for Integer DFO

Proposed by Audet & Dennis (2001):

- User-defined discrete neighborhood

Pattern-Search Algorithm for Integer DFO

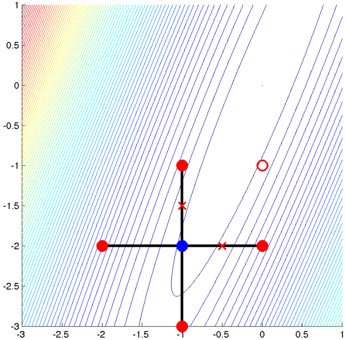
Proposed by Audet & Dennis (2001):

- User-defined discrete neighborhood
- Declare “mesh-isolated minimizer” if no local improvement

Pattern-Search Algorithm for Integer DFO

Proposed by Audet & Dennis (2001):

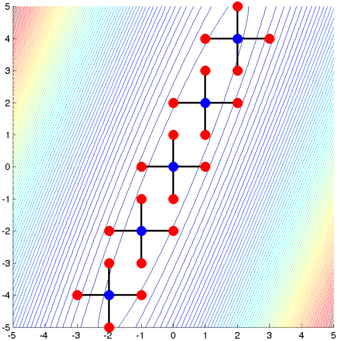
- User-defined discrete neighborhood
- Declare "mesh-isolated minimizer" if no local improvement



Pattern-Search Algorithm for Integer DFO

Proposed by Audet & Dennis (2001):

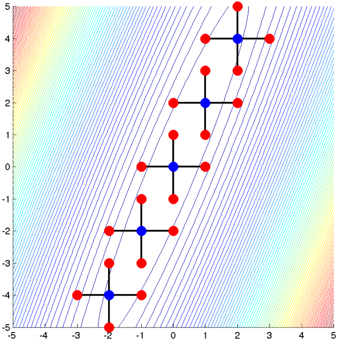
- User-defined discrete neighborhood
- Declare "mesh-isolated minimizer" if no local improvement



Pattern-Search Algorithm for Integer DFO

Proposed by Audet & Dennis (2001):

- User-defined discrete neighborhood
- Declare "mesh-isolated minimizer" if no local improvement



- Any $(y_1, y_2) \in \mathbb{Z}^2$ with $2y_1 = y_2$ is optimal

Discussion

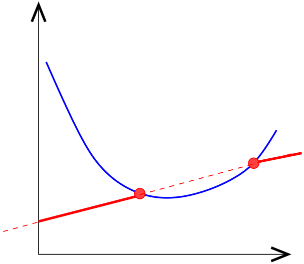
Open problem

Can we guarantee a minimizer of a convex $f(x)$ when x is integer?

1D

$$\underset{x}{\text{minimize}} \;\; f(x), \quad \text{subject to } x \in \mathbb{Z}^n$$

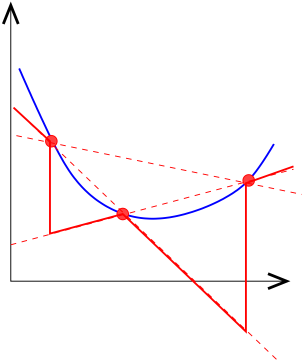
and assume $f(x)$ convex



1D

$$\underset{x}{\text{minimize}} \ f(x), \quad \text{subject to } x \in \mathbb{Z}^n$$

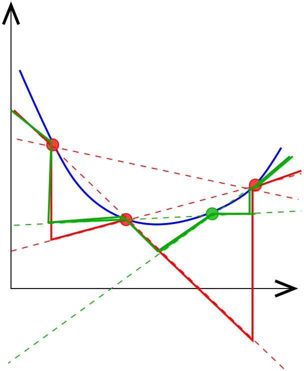
and assume $f(x)$ convex



1D

$$\underset{x}{\text{minimize}} \; f(x), \quad \text{subject to } x \in \mathbb{Z}^n$$

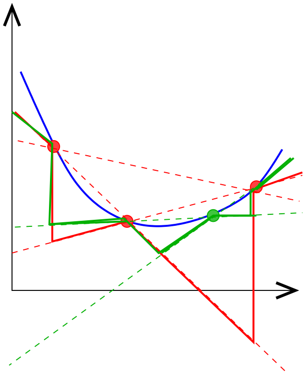
and assume $f(x)$ convex



1D

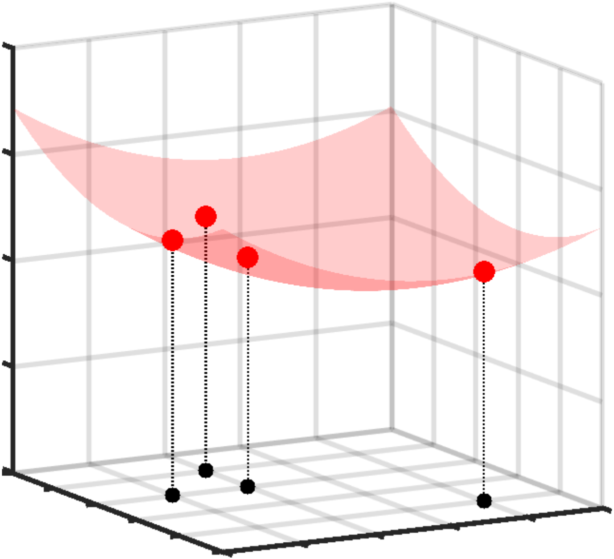
$$\underset{x}{\text{minimize}}\; f(x), \quad \text{subject to } x \in \mathbb{Z}^n$$

and assume $f(x)$ convex

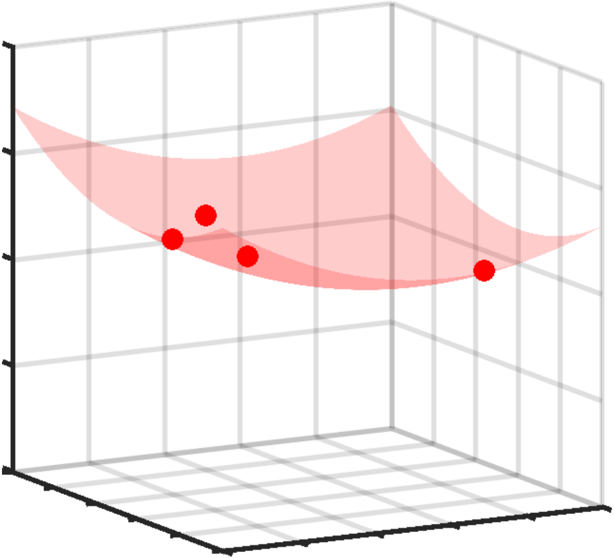


... underestimator for convex, integer DFO!

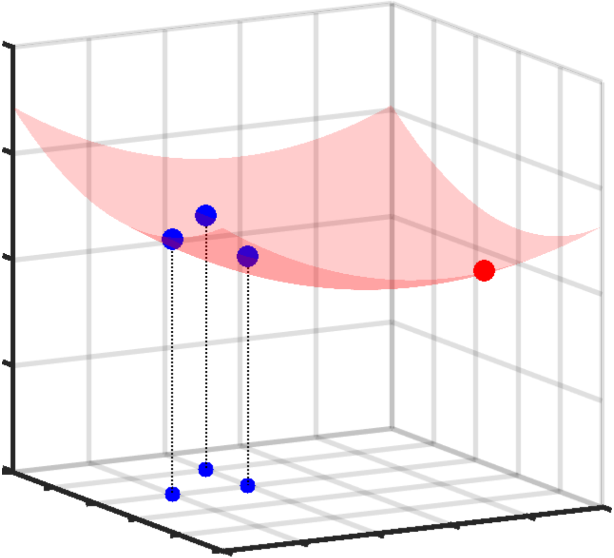
2D



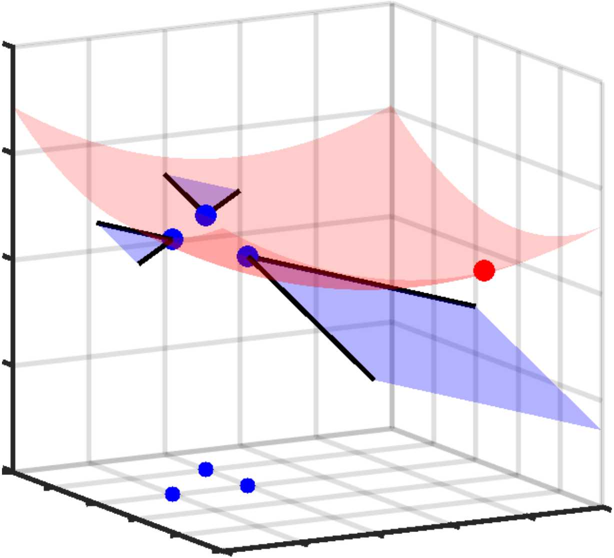
2D



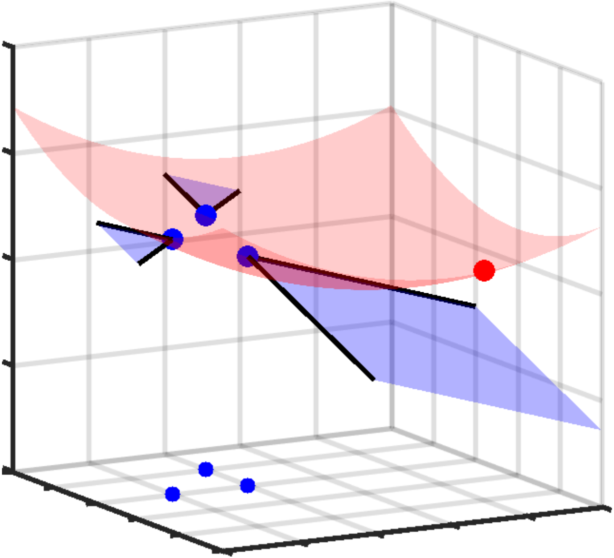
2D



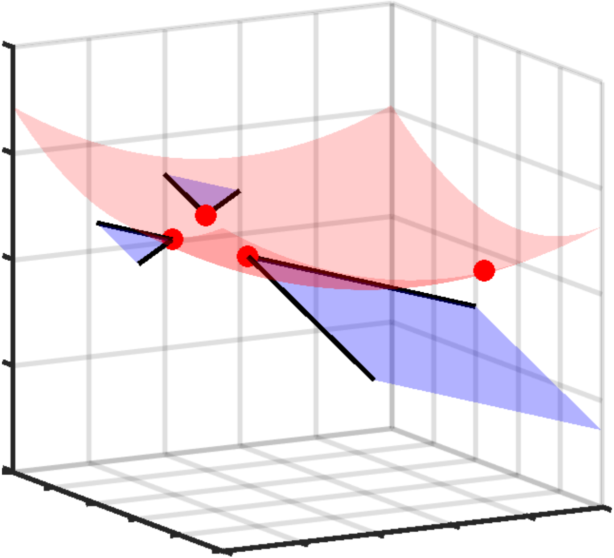
2D



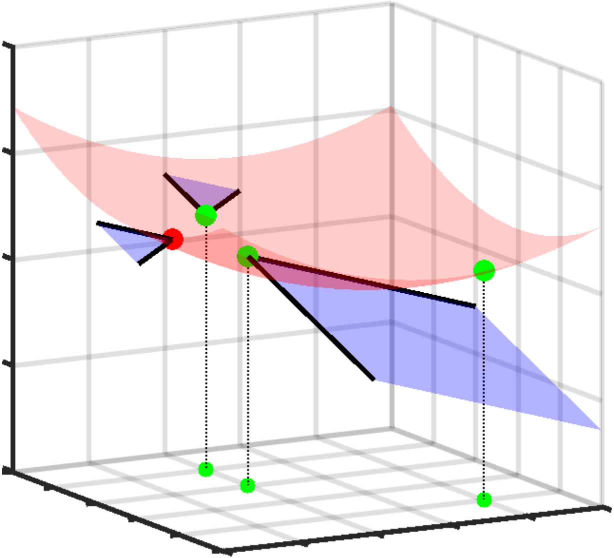
2D



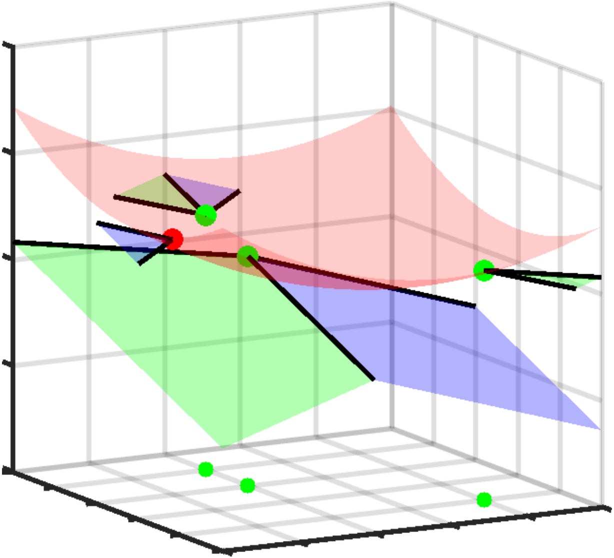
2D



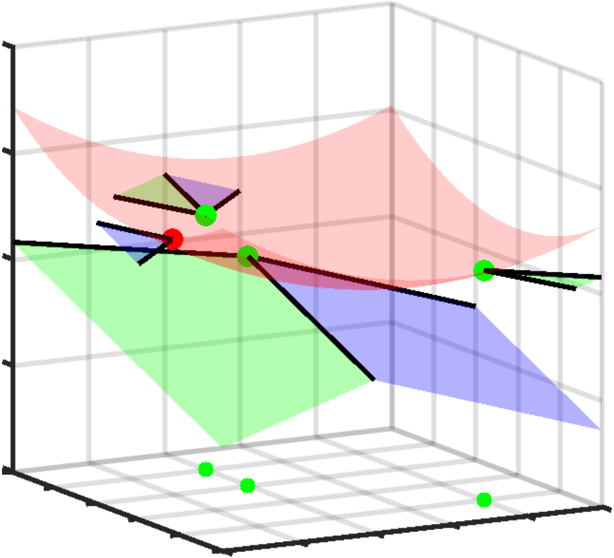
2D



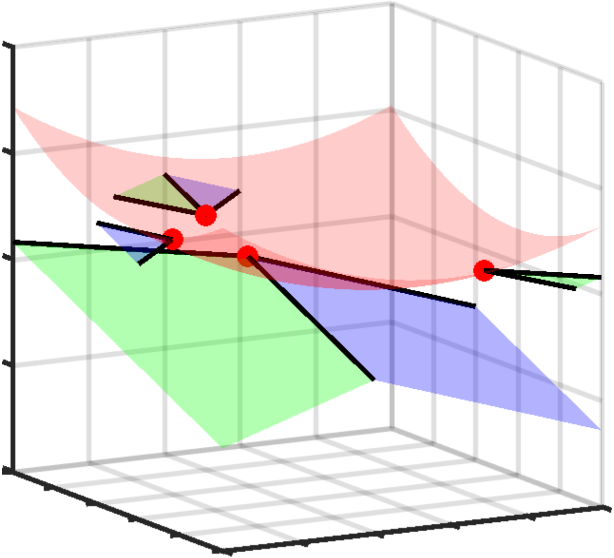
2D



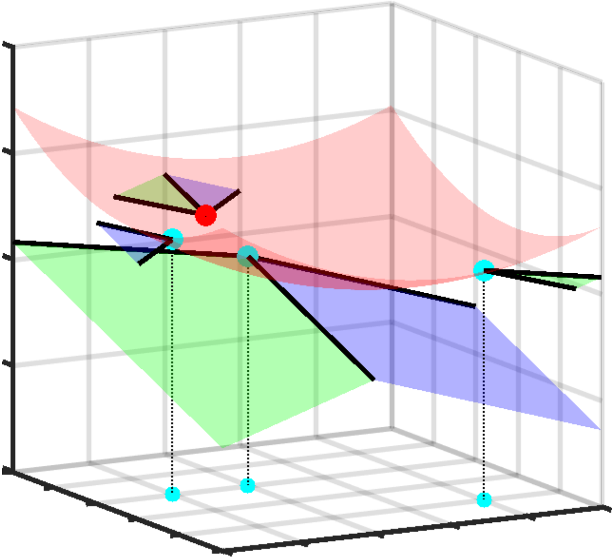
2D



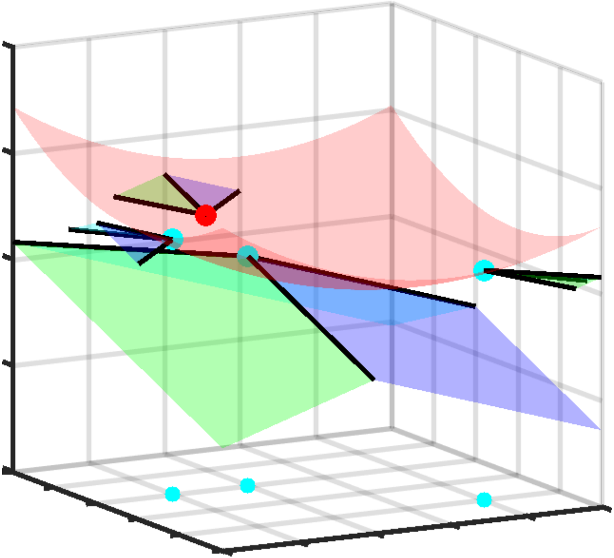
2D



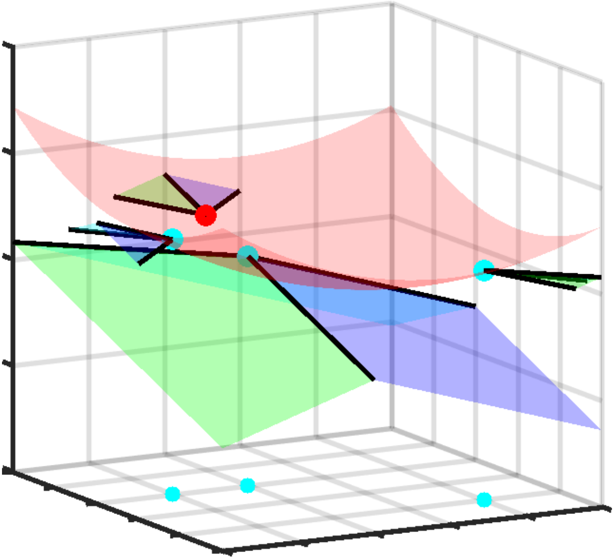
2D



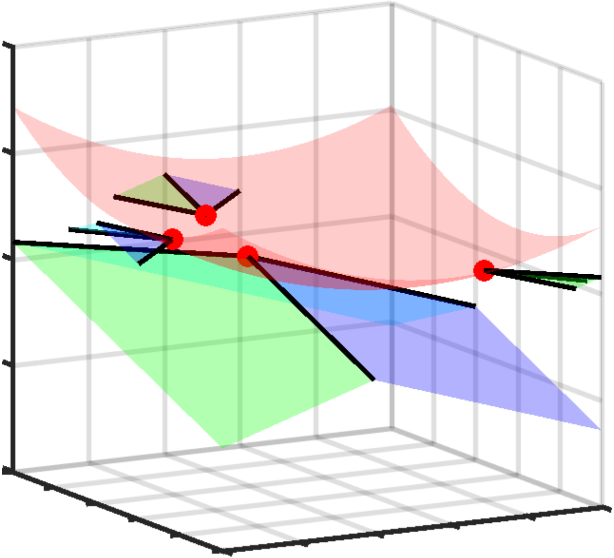
2D



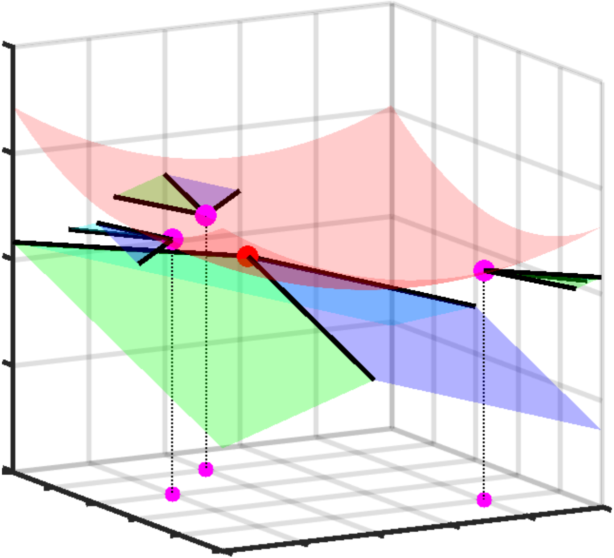
2D



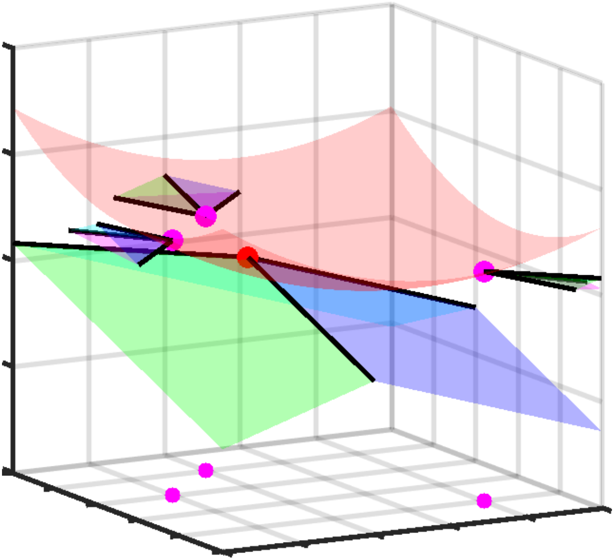
2D



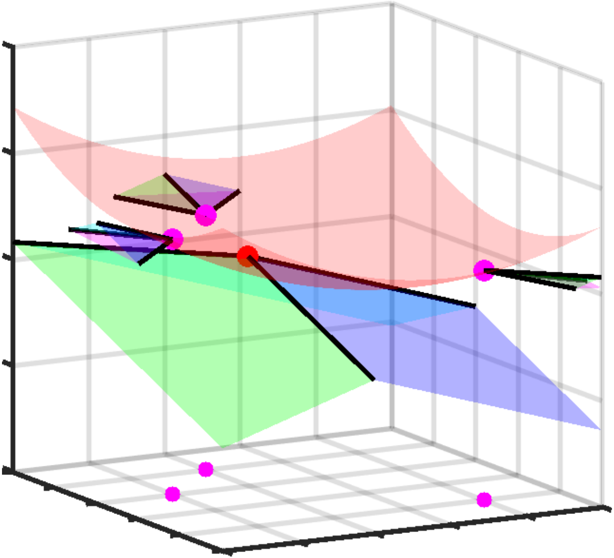
2D



2D



2D



Underestimating f

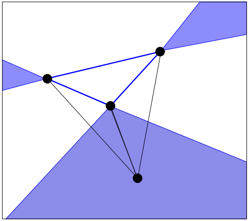
Formulate piecewise underestimator as MILP

- Interpolation points: $X := \{x^i \in \mathbb{Z}^n\}$, $|X| \geq n+1$
- Function values: $f^i := f(x^i)$ for $x^i \in X$
- $\boldsymbol{i} := (i_1, \ldots, i_{n+1})$ multi-index for $n+1$ distinct $i_j \in \boldsymbol{i}$ with $1 \leq i_1 < \ldots, i_{n+1} \leq |X|$

Interpolation Cuts

For $X^{\boldsymbol{i}} := \{x^{i_j} : i_j \in \boldsymbol{i}\}$ obtain cut $(c^{\boldsymbol{i}})^T x + b^{\boldsymbol{i}}$
... only valid in cones ... by solving linear system:

$$\begin{bmatrix} X^{\boldsymbol{i}} & e \end{bmatrix} \begin{bmatrix} c^{\boldsymbol{i}} \\ b^{\boldsymbol{i}} \end{bmatrix} = f^{\boldsymbol{i}},$$



Underestimating f

Lemma: Underestimating Property

$f(x)$ convex and $X^{\boldsymbol{i}} = \{x^{i_1}, \ldots, x^{i_{n+1}}\}$ poised, then if follows that

$$f(x) \geq (c^{\boldsymbol{i}})^T x + b^{\boldsymbol{i}}, \qquad \forall\, x \in U^{\boldsymbol{i}} := \bigcup_{j=1}^{n+1} \mathrm{cone}\left(x^{i_j} - X^{\boldsymbol{i}}\right),$$

where $\mathrm{cone}\left(x^{i_j} - X^{\boldsymbol{i}}\right)$ is the cone with vertex $x^{i_j} \in X^{\boldsymbol{i}}$ & rays $x^{i_j} - x^{i_l}$:

Solving the subproblem - MILP formulation

Modeling membership in cones (using $\eta \geq f(x)$ epigraph trick)

- Binary $z^{i_j} = 1$ if and only if $x \in \operatorname{cone}\left\{x^{i_j} - X^{\boldsymbol{i}}\right\},$ for $i_j \in \boldsymbol{i}$

Solving the subproblem - MILP formulation

Modeling membership in cones (using $\eta \geq f(x)$ epigraph trick)

- Binary $z^{i_j} = 1$ if and only if $x \in \operatorname{cone}\left\{x^{i_j} - X^{\boldsymbol{i}}\right\}$, for $i_j \in \boldsymbol{i}$
- Cut $\eta \geq (c^{\boldsymbol{i}})^T x + b^{\boldsymbol{i}} - M_{\boldsymbol{i}}(1 - \sum_{j=1}^{n+1} z^{i_j})$ for big-$M_{\boldsymbol{i}} > 0$

Solving the subproblem - MILP formulation

Modeling membership in cones (using $\eta \geq f(x)$ epigraph trick)

- Binary $z^{i_j} = 1$ if and only if $x \in \mathrm{cone}\left\{x^{i_j} - X^{\boldsymbol{i}}\right\}$, for $i_j \in \boldsymbol{i}$
- Cut $\eta \geq (c^{\boldsymbol{i}})^T x + b^{\boldsymbol{i}} - M_{\boldsymbol{i}}(1 - \sum_{j=1}^{n+1} z^{i_j})$ for big-$M_{\boldsymbol{i}} > 0$
- SOS-1 constraint: at most one cone, $z^{i_j} \leq 1$, active

Solving the subproblem - MILP formulation

Modeling membership in cones (using $\eta \geq f(x)$ epigraph trick)

- Binary $z^{i_j} = 1$ if and only if $x \in \operatorname{cone}\left\{x^{i_j} - X^{\boldsymbol{i}}\right\}$, for $i_j \in \boldsymbol{i}$
- Cut $\eta \geq (c^{\boldsymbol{i}})^T x + b^{\boldsymbol{i}} - M_{\boldsymbol{i}}(1 - \sum_{j=1}^{n+1} z^{i_j})$ for big-$M_{\boldsymbol{i}} > 0$
- SOS-1 constraint: at most one cone, $z^{i_j} \leq 1$, active
- Any point x is linear combination of extreme rays ($W(X)$ set of all poised subsets)

$$x = x^{i_j} + \sum_{\substack{l=1,\\ l\neq j}}^{n+1} \lambda_l^{i_j}(x^{i_j} - x^{i_l}), \quad \forall i_j \in \boldsymbol{i},\ \forall \boldsymbol{i} \in W(X)$$

Solving the subproblem - MILP formulation

Modeling membership in cones (using $\eta \geq f(x)$ epigraph trick)

- Binary $z^{i_j} = 1$ if and only if $x \in \text{cone}\left\{x^{i_j} - X^{\boldsymbol{i}}\right\}$, for $i_j \in \boldsymbol{i}$
- Cut $\eta \geq (c^{\boldsymbol{i}})^T x + b^{\boldsymbol{i}} - M_{\boldsymbol{i}}(1 - \sum_{j=1}^{n+1} z^{i_j})$ for big-$M_{\boldsymbol{i}} > 0$
- SOS-1 constraint: at most one cone, $z^{i_j} \leq 1$, active
- Any point x is linear combination of extreme rays ($W(X)$ set of all poised subsets)

$$x = x^{i_j} + \sum_{\substack{l=1,\\ l\neq j}}^{n+1} \lambda_l^{i_j}(x^{i_j} - x^{i_l}), \quad \forall i_j \in \boldsymbol{i}, \, \forall \boldsymbol{i} \in W(X)$$

- Indicate $x \in \text{cone}\left(x^{i_j} - X^{\boldsymbol{i}}\right)$ by making $\lambda_l^{i_j} \geq -M_\lambda(1 - z^{i_j})$

Solving the subproblem - MILP formulation

Modeling membership in cones (using $\eta \geq f(x)$ epigraph trick)

- Binary $z^{i_j} = 1$ if and only if $x \in \text{cone}\left\{x^{i_j} - X^{\boldsymbol{i}}\right\}$, for $i_j \in \boldsymbol{i}$
- Cut $\eta \geq (c^{\boldsymbol{i}})^T x + b^{\boldsymbol{i}} - M_{\boldsymbol{i}}(1 - \sum_{j=1}^{n+1} z^{i_j})$ for big-$M_{\boldsymbol{i}} > 0$
- SOS-1 constraint: at most one cone, $z^{i_j} \leq 1$, active
- Any point x is linear combination of extreme rays ($W(X)$ set of all poised subsets)

$$x = x^{i_j} + \sum_{\substack{l=1, \\ l \neq j}}^{n+1} \lambda_l^{i_j} (x^{i_j} - x^{i_l}), \quad \forall i_j \in \boldsymbol{i}, \ \forall \boldsymbol{i} \in W(X)$$

- Indicate $x \in \text{cone}\left(x^{i_j} - X^{\boldsymbol{i}}\right)$ by making $\lambda_l^{i_j} \geq -M_\lambda(1 - z^{i_j})$

... models $z^{i_j} = 1 \Rightarrow x \in \text{cone}\left\{x^{i_j} - X^{\boldsymbol{i}}\right\}$ for $i_j \in \boldsymbol{i}$ ... reverse harder

Solving the subproblem - MILP formulation

Now to show $x \in \text{cone}\left\{x^{i_j} - X^{\boldsymbol{i}}\right\} \Rightarrow z^{i_j} = 1$

- $\lambda_l^{i_j} \geq 0$ implies $w_l^{i_j} = 1$ and $\lambda_l^{i_j} \leq -\epsilon_\lambda$ implies $w_l^{i_j} = 0$

  $$\lambda_l^{i_j} \leq -\epsilon_\lambda + M_\lambda w_l^{i_j} \qquad \text{...tiny-}\epsilon \text{ and big-}M$$

  ... can choose optimal tiny-ϵ, big-M by solving LPs
- At least one w variables is zero if corresponding z is zero:

  $$n z^{i_j} \leq \sum_{l=1,\, l \neq j}^{n+1} w_l^{i_j} \leq n - 1 + z^{i_j}$$

MILP subproblem model

- Check $\binom{|X|}{n+1}$ poised sets at iteration k
- Formulation needs $\mathcal{O}(n^2)$ binary variables per poised set

An Alternative Master Problem

Challenges of MILP Master model

- MILP model exponential in number of interpolation points
- MILP representation is very weak: uses big-M and tiny-ϵ

$\Rightarrow$ Commercial solvers cannot solve large instances

An Alternative Master Problem

Challenges of MILP Master model

- MILP model exponential in number of interpolation points
- MILP representation is very weak: uses big-M and tiny-ϵ

$\Rightarrow$ Commercial solvers cannot solve large instances

Replacing CPLEX Solve by Look-Up Table

- Key idea: work in space of original integers, $x \in \mathbb{Z}^n$ (no additional variables or constraints)
- Replace MILP by look-up-table of underestimator
- Update look-up-table when new points (and therefore new cuts) are available

An Alternative Master Problem

Challenges of MILP Master model

- MILP model exponential in number of interpolation points
- MILP representation is very weak: uses big-M and tiny-ϵ

$\Rightarrow$ Commercial solvers cannot solve large instances

Replacing CPLEX Solve by Look-Up Table

- Key idea: work in space of original integers, $x \in \mathbb{Z}^n$ (no additional variables or constraints)
- Replace MILP by look-up-table of underestimator
- Update look-up-table when new points (and therefore new cuts) are available

Dense/small linear algebra solves $\Rightarrow$ Fast

An Alternative Master Problem

Challenges of MILP Master model

- MILP model exponential in number of interpolation points
- MILP representation is very weak: uses big-M and tiny-ϵ

$\Rightarrow$ Commercial solvers cannot solve large instances

Replacing CPLEX Solve by Look-Up Table

- Key idea: work in space of original integers, $x \in \mathbb{Z}^n$ (no additional variables or constraints)
- Replace MILP by look-up-table of underestimator
- Update look-up-table when new points (and therefore new cuts) are available

Dense/small linear algebra solves $\Rightarrow$ Fast . . . but not fast enough

```
Algorithm 1: Look up algorithm
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
```

```
Algorithm 1: Look up algorithm
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
```

Algorithm 1: Look up algorithm

```
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n+1$ points in $X\}$
```

```
Algorithm 1: Look up algorithm
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n+1$ points in $X\}$
    for $i \in C$ do
        if $X^{\boldsymbol{i}}$ is poised then
```

Algorithm 1: Look up algorithm

```
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n+1$ points in $X\}$
    for $\boldsymbol{i} \in C$ do

        if $X^{\boldsymbol{i}}$ is poised                then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$
```

```
Algorithm 1: Look up algorithm
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n+1$ points in $X\}$
    for $\boldsymbol{i} \in C$ do
        if $X^{\boldsymbol{i}}$ is poised then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$
            Update $\eta$
```

```
Algorithm 1: Look up algorithm
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n+1$ points in $X\}$
    for $\boldsymbol{i} \in C$ do
        if $X^{\boldsymbol{i}}$ is poised then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$
            Update $\eta$
    Add $x^k = \arg\min_{x \in \Omega} \eta$ and update $\bar{f}$
```

```
Algorithm 1: Look up algorithm
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n+1$ points in $X\}$
    for $\boldsymbol{i} \in C$ do
        $[Q, R] = \mathrm{qr}([e\ X^{\boldsymbol{i}}])$
        if $X^{\boldsymbol{i}}$ is poised            then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$
            Update $\eta$
    Add $x^k = \arg\min_{x \in \Omega} \eta$ and update $\bar{f}$
```

```
Algorithm 1: Look up algorithm
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
        $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n+1$ points in $X\}$
    for $\boldsymbol{i} \in C$ do
        $[Q, R] = \mathrm{qr}([e\ X^{\boldsymbol{i}}])$
        if $X^{\boldsymbol{i}}$ is poised using $R$ then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$
            Update $\eta$
    Add $x^k = \arg\min_{x \in \Omega} \eta$ and update $\bar{f}$
```

Algorithm 1: Look up algorithm

```
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n+1$ points in $X\}$
    for $\boldsymbol{i} \in C$ do
        $[Q, R] = \mathrm{qr}([e\ X^{\boldsymbol{i}}])$
        if $X^{\boldsymbol{i}}$ is poised using $R$ then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$ using $Q$
            Update $\eta$
    Add $x^k = \arg\min_{x \in \Omega} \eta$ and update $\bar{f}$
```

```
Algorithm 1: Look up algorithm
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n+1$ points in $X\}$
    for $\boldsymbol{i} \in C$ do
        $[Q, R] = \mathrm{qr}([e\ X^{\boldsymbol{i}}])$
        if $X^{\boldsymbol{i}}$ is poised using $R$ then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$ using $Q$
            Update $\eta$ using $Q$
    Add $x^k = \arg\min_{x \in \Omega} \eta$ and update $\bar{f}$
```

```
Algorithm 1: Look up algorithm
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n+1$ points in $X\}$
    for $\boldsymbol{i} \in C$ do
        $[Q, R] = \mathrm{qr}([e\ X^{\boldsymbol{i}}])$
        if $X^{\boldsymbol{i}}$ is poised using $R$ then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$ using $Q$
            Update $\eta$ using $Q$ where $\eta < \bar{f}$
    Add $x^k = \arg\min_{x \in \Omega} \eta$ and update $\bar{f}$
```

```
Algorithm 1: Look up algorithm
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n+1$ points in $X\}$
    for $\boldsymbol{i} \in C$ do
        $[Q, R] = \mathrm{qr}([e\ X^{\boldsymbol{i}}])$
        if $X^{\boldsymbol{i}}$ is poised using $R$ then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$ using $Q$
            Update $\eta$ using $Q$ where $\eta < \bar{f}$
    Add $x^k = \arg\min_{x \in \Omega, \eta < \bar{f}} \eta$ and update $\bar{f}$
```

```
Algorithm 1: Look up algorithm
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n$ points in $X\} \otimes \{x^k\}$
    for $\boldsymbol{i} \in C$ do
        $[Q, R] = \mathrm{qr}([e\ X^{\boldsymbol{i}}])$
        if $X^{\boldsymbol{i}}$ is poised using $R$ then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$ using $Q$
            Update $\eta$ using $Q$ where $\eta < \bar{f}$
    Add $x^k = \arg\min_{x \in \Omega, \eta < \bar{f}} \eta$ and update $\bar{f}$
```

```
Algorithm 1: Look up algorithm
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    $C = \{$subsets of $n+1$ useful points in $X\}$
    for $\boldsymbol{i} \in C$ do
        $[Q, R] = \mathrm{qr}([e\ X^{\boldsymbol{i}}])$
        if $X^{\boldsymbol{i}}$ is poised using $R$ then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$ using $Q$
            Update $\eta$ using $Q$ where $\eta < \bar{f}$
    Add $x^k = \arg\min_{x \in \Omega, \eta < \bar{f}} \eta$ and update $\bar{f}$
```

Algorithm 1: Look up algorithm

```
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    Generate a sufficient set $C$ of subsets of $n+1$ points
    for $\boldsymbol{i} \in C$ do
        $[Q, R] = \mathrm{qr}([e\ X^{\boldsymbol{i}}])$
        if $X^{\boldsymbol{i}}$ is poised using $R$ then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$ using $Q$
            Update $\eta$ using $Q$ where $\eta < \bar{f}$
    Add $x^k = \arg\min_{x \in \Omega, \eta < \bar{f}} \eta$ and update $\bar{f}$
```

Algorithm 1: Look up algorithm

```
input: Lower bound $\eta$ for each point in $\Omega$; Points $X$ with
       $|X| \geq n+1$; $\bar{f} = \min_{x_i \in X} f(x_i)$
while $\bar{f} > \min \eta$ do
    Generate a sufficient set $C$ of subsets of $n+1$ points
    for $\boldsymbol{i} \in C$ do
        $[Q, R] = \mathrm{qr}([e\ X^{\boldsymbol{i}}])$
        if $X^{\boldsymbol{i}}$ is poised using $R$ then
            Find points in $\Omega$ in one of the cones of $X^{\boldsymbol{i}}$ using $Q$
            Update $\eta$ using $Q$ where $\eta < \bar{f}$
    Add $x^k = \arg\min_{x \in \Omega, \eta < \bar{f}} \eta$ and update $\bar{f}$
```

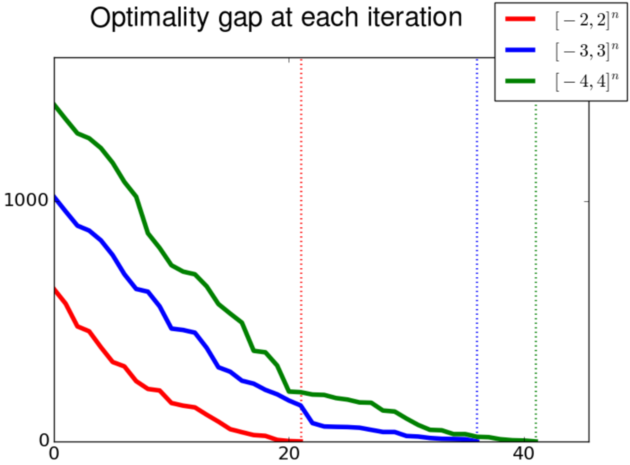
Open problem

Given a set of points X on the integer lattice, is there a way to generate all subsets of size $n+1$ without another in the interior?

Results Abhishek Function

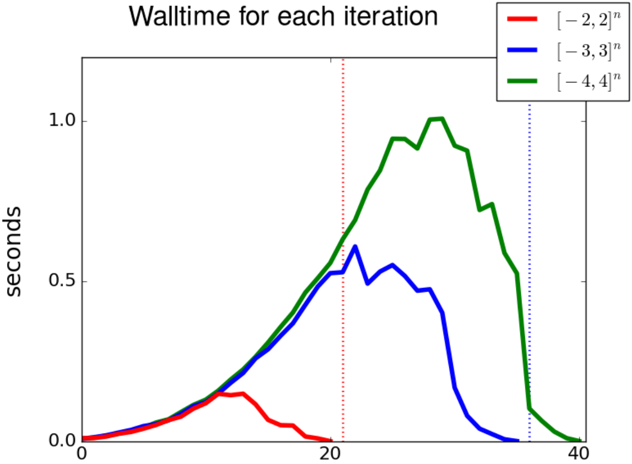
Convex quadratic for which pattern-search failed ...

$n = 3$



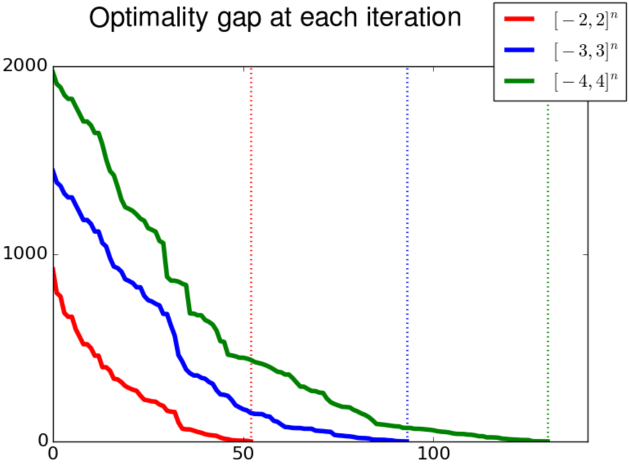
Results Abhishek Function

Convex quadratic for which pattern-search failed ...
$n = 3$



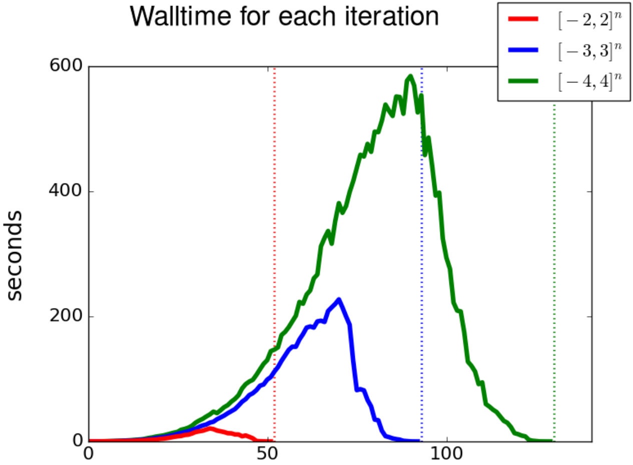
Results Abhishek Function

Convex quadratic for which pattern-search failed ...
$n = 4$



Results Abhishek Function

Convex quadratic for which pattern-search failed ...
$n = 4$



Future

- We likely aren't using convexity as much as we possibly can.
- How to certify (local) optimality when f is nonconvex?